



Expertise  
and insight  
for the future

Andriy Bakshalov

# Similarity Analysis Between Groups of Construction Objects in BIM Application

Metropolia University of Applied Sciences

Master of Engineering

Information Technology

Master's Thesis

10 March 2020

## PREFACE

I would like to express my gratitude to Mr. Ville Jääskeläinen Head of Master's Degree in Metropolia University of Applied Sciences for supervising this thesis and for all the support and advice on a long journey to the finish line. Also, I would like to express my gratitude to Mr. Henri Pitkänen, Software Manager at Trimble Inc for finding me such an interesting topic to research and help in understanding different phases of construction processes and connection design as well as offering help during the implementation. During research this thesis showed me different sides of a construction process and many of the challenges that civil engineers face every day which, probably, I would never discover on my own.

A big thank you to my friends, my fluffy dog and my girlfriend for a continuous support and motivation to achieve the finish line of the project and my studies.

The last, but not least a gratitude to my parents that were always motivating me to apply for master's studies as early as possible and not to leave it as a "future plan" that might be accomplished someday.

Helsinki, 09.03.2020

Andriy Bakshalov.

Author Title	Andriy Bakshalov Similarity Analysis Between Groups of Construction Objects in BIM Application
Number of Pages Date	50 pages 09 March 2020
Degree	Master of Engineering
Degree Programme	Information Technology
Instructor(s)	Henri Pitkänen, Software Manager Ville Jääskeläinen, Supervisor
<p>This thesis explores possibilities of using similarity to create a new connection between a set of construction objects from previously finished projects in a Building Information Modelling software called Tekla Structures.</p> <p>For all the construction processes a connection is one of the most important things that keep different construction objects connected. There are different existing solutions provided by the software, but the benefits to users are limited and a lot of time during project development is spent on designing connections. To find a better solution that would save time and effort for the users this study focuses on finding out if similarity from the previous projects could benefit users in their future projects.</p> <p>To find answer to this problem a functional prototype was developed that would collect and analyse existing project connections and using different techniques to find out what would be a best suggestion for a new connection between selected set of construction objects in Tekla Structures software.</p> <p>The technical implementation of the prototype was written in C# and C++ programming languages that was inbuilt into Tekla Structures to enable existing code to operate with models and extract information using API written in C#. The creation of a new suggested connection was written in C++ as a native development language of Tekla Structures.</p> <p>The results of this study show how the developed prototype performed with the selected model used for the testing. Overall outcomes were satisfactory, showing promising results, but at the same time not all the test cases passed as expected, leaving possibility for a future improvement and development that are described in conclusions.</p>	
Keywords	Building Information Modelling, Tekla Structures, Construction

## Contents

Preface

Abstract

List of Abbreviations

1	Introduction	1
1.1	Problem Statement	5
1.2	Structure of Thesis	5
2	Current State Analysis	7
2.1	Company	7
2.2	Computer Aided Design and Building Information Modelling	7
2.3	Tekla Structures	8
2.4	Connections	9
2.5	Autoconnection	11
2.6	Ongoing Research	12
3	Methods Used During Implementation	14
3.1	Application Programming Interface	14
3.2	Feature Selection	15
3.3	Clustering	17
3.3.1	Connectivity-based Clustering	17
3.3.2	Distribution-based Clustering	19
3.3.3	Density-based Clustering	19
3.3.4	Centroid-based Clustering	20
3.4	Similarity Methods	21
3.5	Alternative Methods	25
3.5.1	Decision Tree	26
3.5.2	Neural Networks	28
3.6	Project Plan	29
4	Implementation	31
4.1	Case Study Model	31
4.2	Alternative Methods Approach	36
5	Results and Analysis	38

5.1	Case 1	38
5.2	Case 2	40
5.3	Case 3	42
5.4	Case 4	44
5.5	Case 5	46
6	Conclusions	49

## References

## List of Abbreviations

CAD – Computer Aided Design

BIM – Building Information Modelling

TS – Tekla Structures

API – Application Programming Interface

## 1 Introduction

From ancient times, building a safe place was a vital for a human survival. Starting from stone age, humans have used different techniques and tools that were invented to construct safe places for living and to protect from the wild animals and environmental hazards [1]. A constant population growth was facilitating faster evolution in construction techniques and tools. At the same time physics and mathematics were in a constant development that produced influence on construction design. Due to different discoveries in science fields, construction projects were becoming more and more complex that ended up with a decision to have specialized personnel to verify project plans. These professionals are called civil engineers today.

Canadian society for civil engineering determines that civil engineering is the second oldest engineering discipline after military [2]. In the beginning, the role included both an architect and a civil engineer, making the same person to design and provide different mathematical and physical calculation for a construction project. A great definition of a civil engineer is given by Institution of Civil Engineering as:

“the art of directing the great sources of power in nature for the use and convenience of man, as the means of production and of traffic in states, both for external and internal trade, as applied in the construction of roads, bridges, aqueducts, canals, river navigation and docks for internal intercourse and exchange, and in the construction of ports, harbours, moles, breakwaters and lighthouses, and in the art of navigation by artificial power for the purposes of commerce, and in the construction and application of machinery, and in the drainage of cities and towns.”

When constructions become more complex, a civil engineering role was completely separated from architects. It is a very generic term to describe a person that is specialized in construction workflow, and nowadays it is often split into sub-disciplines such as: construction, structural, etc.

In a usual workflow for designing a new construction project, there is first an architect who presents the planned end result and then it's a job of specialized engineers to find a way how to model and construct proposed structures. Sometimes, architects design a very futuristic and risky structures that might introduce a great challenge to be built using existing technologies and tools. With a more and more demanding building requirements and more dangerous projects to be accomplished by architects, civil engineering became

an important part of a new construction project that verified different risks and planning to lead the project until a successful end.

A typical tool during design and planning of a new construction includes multiple technical drawings for each construction object to be fabricated. A set of construction objects in a single drawing is called an assembly drawing that gives instructions on how multiple objects are connected and positioned together. For example, a simple technical drawing is shown in following Figure 1.

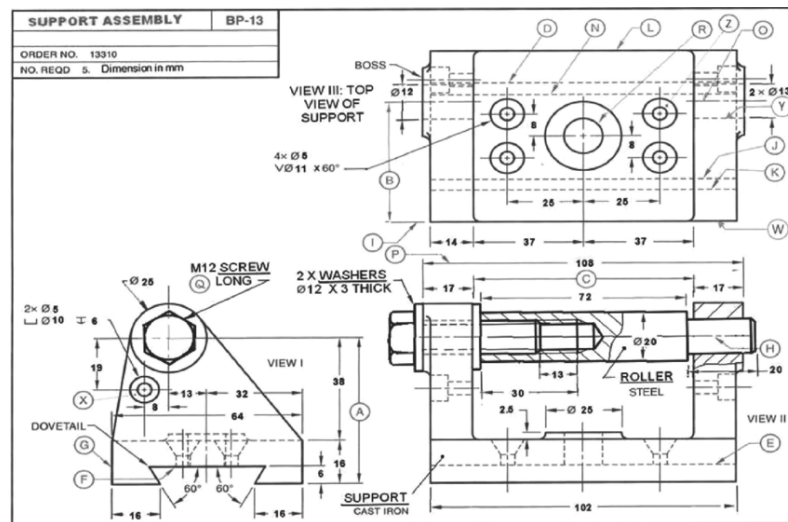


Figure 1. Technical drawing of a part.

In this figure, there is all the information needed to create a certain detailed part. Usually they contain different views, such as, top, front and side views that provide required information to draw a 3D visualization. Other obligatory information are dimensions such as: length, width, height, size of bolt holes and weld size to use, as well as, tolerances to follow. There can be annotations and some instructions written to ease the fabrication if some of the details are not simple to understand. One more important feature is that drawings should provide information about different country standards that the designer was using, as different countries have their own fabrication standards and metrics.

After computers were invented and their computing power was increased dramatically, more technological possibilities came towards construction industry. This enabled a development of a Computer Aided Design [3] (CAD) software that made it possible to design in a digital form different construction objects and produce technical drawings of them. This help from software made designing a less time-consuming process, as well



as later, it was possible to share digital models using Internet. Nowadays a construction project is not carried by one or few individuals no matter how experienced they are. Usually it involves different companies and requires hundreds of professionals from different areas. With the help of CAD, it was possible to share models between different specialists or companies to review and correct different design decisions. In the recent years, Building Information Modelling [4] (BIM) started to appear and take over CAD solutions as it provides users with more advantages such as:

- Improved collaboration between information suppliers and users in the facility business.
- More accurate fundamental information to support decision making.
- A standard way of storing information so that it survives over of time.
- A database that can be used to support other business processes.

More detailed information about computer aided design and building information modelling is given in the next chapter.

Given this introduction of construction industry, this thesis concentrates on designing connections between groups of construction objects. Connection itself represents a separate detail to connect two or more construction objects. They could be fixed (connected parts cannot move) and not fixed (connected objects have some movement). Usually it is hard to see connections inside or outside of a building as they are covered by design-oriented covers. As an example, Figure 2 shows a basic connection between a beam and a column. This photo was taken by the thesis author in Ruoholahti, Helsinki. It shows an example of connection usage in a real world. The figure shows a connection between a beam and a column using four bolts and two shear plates that are welded to each larger part and then bolted to each other.



Figure 2. A connection example in a building at Ruoholahti.

To design one connection, it might take from ten minutes and up to months of work, if not longer. There are many ways to design connections and usually they are not trivial to do. For this reason, specific software solutions exist and experts in connection design should apply different analysis methods to know what kind of a connection to use and how many bolts should be used. This is where the case study software Tekla Structures helps different civil engineers and connection design specialist in their jobs. The mentioned software has its own catalogue of different connections and settings to make a perfect solution. What more, it has a feature called Autoconnection which helps to make connection design more automatic. Usually same type of connections repeats during the project and it is much easier to set up rules for auto-connect rather than create them from scratch over and over [5].

As it was mentioned previously, progress never stops. For this reason, this thesis research was provided by Trimble Inc. for their software Tekla Structures, where the aim is to find a better way to create connections based on previous projects and to increase productivity.

## 1.1 Problem Statement

With a better computing power, machine learning [6] has become an important tool in modern life. More and more data from different fields is used to make processes more automated. In this case, Tekla Structures users have finished models that contain a lot of useful information to be studied. For this research project, connections will be studied, and the major questions of this study are:

*Is it possible to find similarities between connections? If so, is it possible to apply these similarities to automatically create new connections to a new model based on previous projects?*

These are quite wide questions in a field of construction and to narrow down the scope of the project, only two user models are used for this work. This means that only a limited amount of connections, that are present in the given models, are included in this study. The objective of the thesis is to implement a tool that is able to gather and process existing data and then create a new similar connection between parts.

The outcome of the thesis is an extension tool for Tekla Structures software that can create a new connection between a selected group of construction objects with high accuracy rate.

## 1.2 Structure of Thesis

This thesis is divided into six different chapters. In the first chapter a short introduction is given to the reader to better understand what a construction industry is, who are civil engineers and what kind of tools they use in their daily work. Also, a short introduction to connection and autoconnection functionalities is given. The second chapter gives detailed information about the case company, the case study software, connections and the autoconnection feature. Following, the situation before starting this work is explained. The current ongoing project in the case company, that is connected to this research, is also covered.

Third chapter explains what methods were used during the implementation and gives technical understanding on how the implementation part was done, ending with a project plan. Following this, fourth chapter tells how this research was carried out using previously explained methods.

To finalize, two last chapters focus on giving a clear view of the results and what was expected in the beginning of the project and what was achieved and a short analysis by a specialist on the achieved results. In the last chapter conclusions of this research project are highlighted, and future steps suggested.

## 2 Current State Analysis

In this chapter, the company, what is Computer Aided Design (CAD) and Building Information Modelling (BIM) and case study Tekla Structures software is briefly explained. Following, what are connections, what is current simplified way of creating them and in the end some information about the current research at the company regarding current project research.

### 2.1 Company

Trimble Inc. [7] is a United States based company that was founded in 1978. Focus of the company was global satellite navigation. With time, the company was growing and decided to start buying smaller companies that would make it grow even bigger. In 2012, Trimble acquires Tekla corporation to expand its influence on the construction industry. From this time, Tekla corporation becomes Trimble Solution Finland with headquarter office at Espoo, Finland with around 400 employees. Focus in headquarters are software solutions for energy distribution, building and structure management. The main software product of Trimble Solution Finland is Tekla Structures which is a building information modelling software that will be described in continuation.

### 2.2 Computer Aided Design and Building Information Modelling

Before explaining study case software, some short information about computer aided design and building information modelling should be introduced before.

With a constant evolution in technology and specifically, in this case, computer technology, new software solutions to help building industry were appearing. Starting from 1960 with more computing power, more possibilities were introduced to software from which the term computer aided design (CAD) is coming from. Using CAD construction engineers could model and design a 2D technical drawing of an object or create a 3D representation of the object and then produce 2D drawings. Usually a finished technical drawing contains information about materials of the modelled object, all the dimensions needed to reproduce the object, accepted tolerances and processes to follow. This information is usually review is done by multiple people and then review by more experienced workers. With introduction of CAD, all this work was a great time saving for a construction project as creation of a technical drawing passed from being done by hand

to a digital format providing faster tools to create, compare and verify before printing the final drawing of the target object.

Computers brought to a construction industry a faster way to work, but the progress didn't stop on that. What if a CAD software would be able to plan the whole construction process? If there would be possibility that it would also calculate the cost of construction at the same time as modelling process is going? That is where building information modelling begins.

The term BIM was introduced since the 1970 and the definition Standard BIM USA organization is the following:

Building Information Modelling (BIM) is a digital representation of physical and functional characteristics of a facility. A BIM is a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life-cycle; defined as existing from earliest conception to demolition.

This means that BIM can provide a whole information of the project. A standard 3D means: width, height and depth, which is what CAD is about. In BIM applications new dimensions as time and cost were introduced. This way, a whole construction project can be modelled in the same software and at the same time provide experts with a more in-depth information of the project, which means that more professionals could be involved in the project using the same model, such as, a design team, architects, detailers, building services engineers, etc. Each party adds their field specific knowledge into the model after which they obtain a complete project which contains from a big overview of the construction planning and costs up to a smallest detail of a well dimensioned technical drawing of a single bolt.

### 2.3 Tekla Structures

Tekla Structures [8] (TS) is one of BIM applications and is the study case software. It is made for structural engineers, designers, fabricators and detailers. Company product description from their web page is:

Tekla Structures is the most advanced structural software for BIM. With it, you can create, combine, manage and distribute accurate multi-material models full of construction information, and manage and communicate the design.

As a BIM software, Tekla Structures makes possible to specify up to the smallest detail in a large construction project. Users can create a 3D view of the project that could be

seen in Figure 3 up to a 2D drawing of the smallest detail using company specific templates and country standards. As well as, there is a multi-user work approach where multiple users can work in parallel on the same project or a model sharing service to visualize models on any device at any time and place. With the help of Tekla Structures construction industry can design anything they would like from a simple detached house to a skyscraper or stadium or even a F1 racing circuit.

Development and improvement of Tekla Structures is an ongoing process and every 6 months there is a major version release and every month there is a service pack releases that fix or improve the quality of the product. So far solution exists for over 25 years and has more than 6 million lines of code and main used development languages are C, C++ and C#.

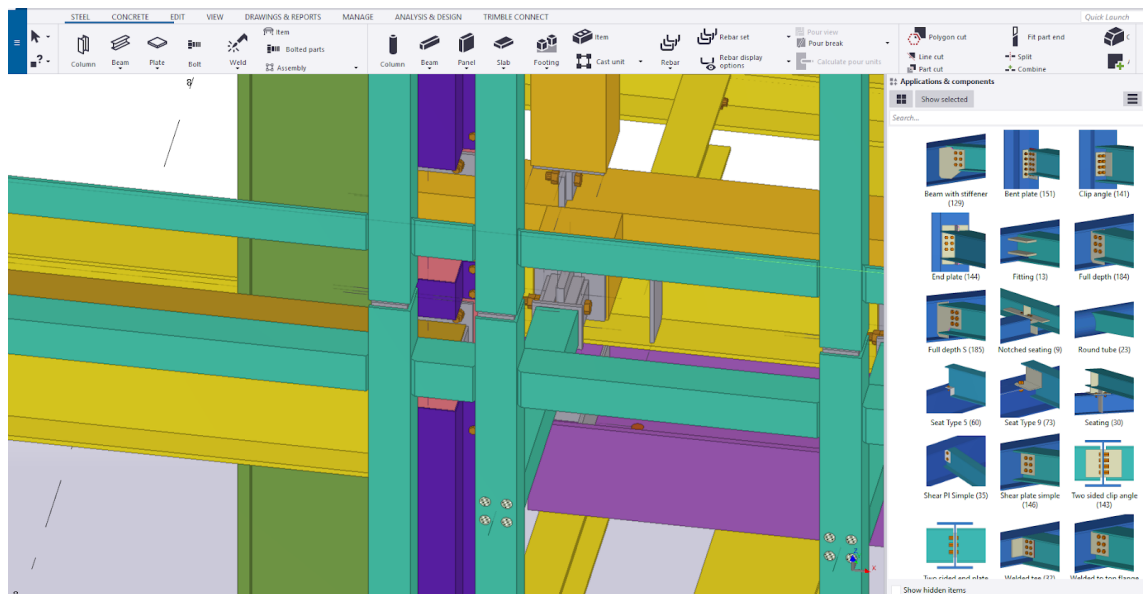


Figure 3. 3D View of Tekla Structures.

## 2.4 Connections

As above mentioned, Tekla Structures gives users the possibility to define models up to smallest possible detail, such as, connection between parts [9]. There is a catalogue of many possible connections between different parts, for example on the Figure 4 there is a screenshot of beam to column connections collection which is just a piece of all the other types.

Using this collection, user can easily find what connection he wants and by selecting parts to connect, it will create the desired connection type. But this is not the lowest detail level of connection.

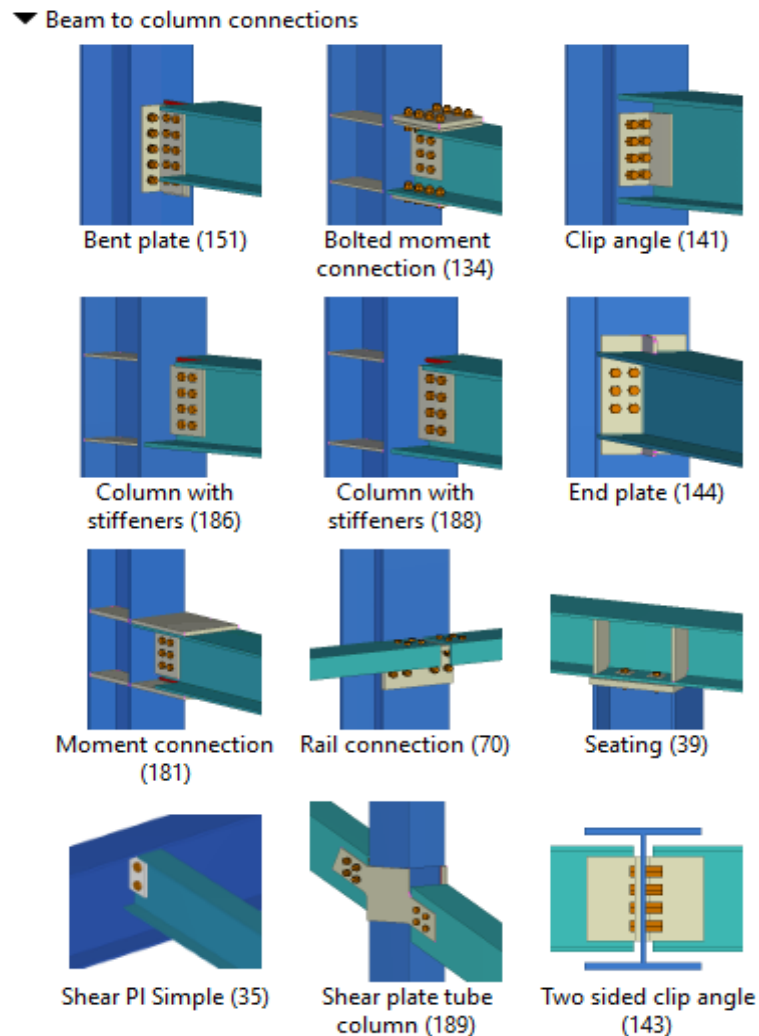


Figure 4. Connection catalogue example for beam to column.

Besides the type, each connection has its own settings and one of the examples is on Figure 5 which shows settings window for an end plate connection. This connection has 11 different tabs of different settings categories. User can specify what type of bolts, welds or additional cuts should be created to achieve the desired outcome.



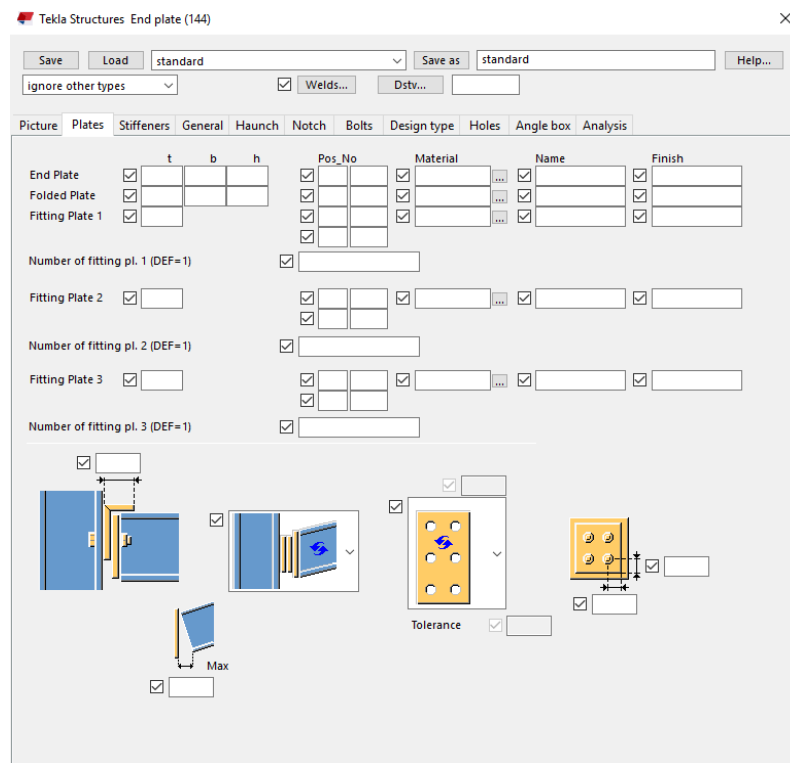


Figure 5. End plate settings window.

All this leads to a very complex structure and a very time-consuming work to define what kind of connection is needed and what configurations should be used after creation. Currently there is a way to simplify this work using an auto-connection feature that is described in continuation.

## 2.5 Autoconnection

As described before, selecting and configuring a connection between set of parts could be rather a time-consuming task, to add more, there are usually thousands of connections on an average sized model. For this reason, autoconnection functionality [10] was implemented. The purpose was to make connection design much easier and less time-consuming process by defining a set of rules by which an algorithm will automatically create a connection and apply predefined setting between objects that a user had selected in the model. In the Figure 6 there is example window of the auto connection setup window. The workflow to create an autoconnection rule is next:

- Select what type of objects will be connection applied to.
- Define properties of objects to follow the rule.

- Setup setting of the connection to follow the rule.

After following these steps, users can reuse the rules and it will automatically create desired connection with specified settings. But, in ideal situation all the connections would follow same rules and settings, usually it is not the case. Usually, between the same parts, connection can differ in position or details, like bolts and welds which make autoconnection unable to adapt to those needs, making user work harder to manually adapt connections to what they need.

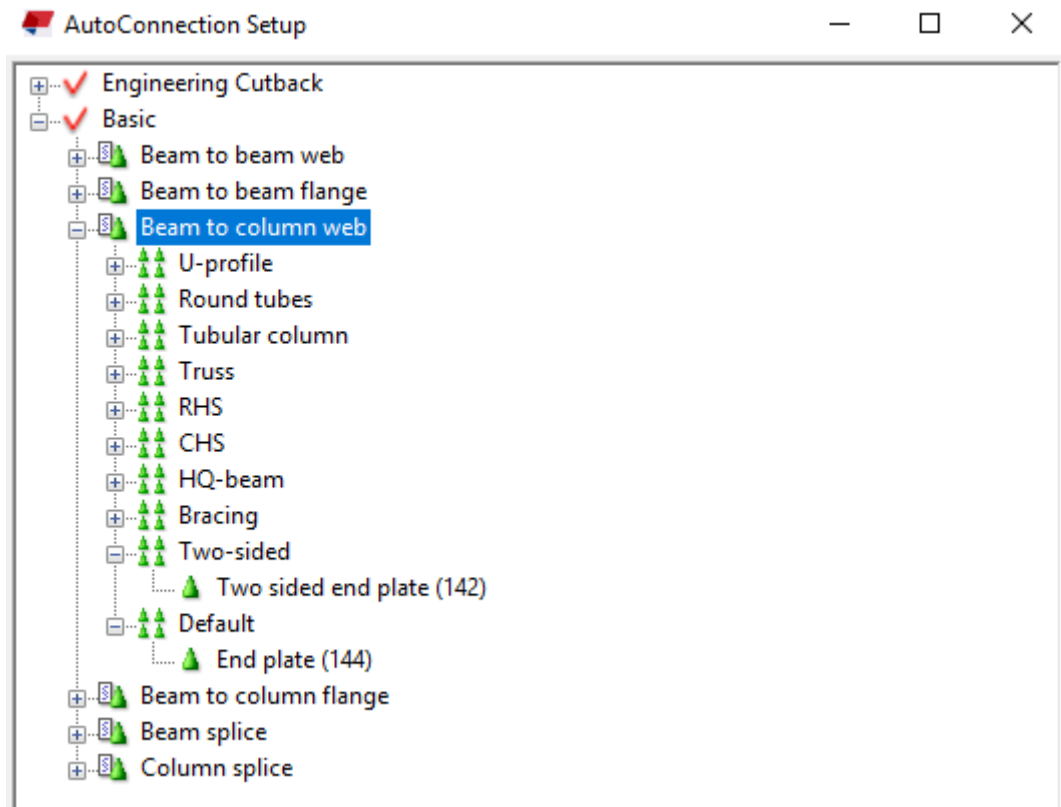


Figure 6. Autoconnection settings window.

## 2.6 Ongoing Research

Before entering in current study plan and more details about the project, it is important to mention ongoing research at the company about autocompletion of set of objects based on the previous modelled similar set of objects.

This work is carried out by analysing existing finished user model and collecting relevant information about set of objects present in it. After data collection, there are different methods that will analyse data and produce sorted and processed information that could be used by implemented algorithms to compare similarities of what is currently selected

by the user and what is present in the processed history database. This research is believed to help the user to autocomplete detailing of different steel and concrete objects. The researched showed very prominent results, as well as, positive user feedback. Unfortunately, it is not as accurate as it expected to be, so it is still under continuous development and improvement work.

This research, about finding similarities and auto-completing sets of construction objects, gave a beginning to the current master thesis research project, to find similarities between modelled connections between sets of construction objects, this way to find out if there is a more efficient way to reduce repetitive work in connection design.

### 3 Methods Used During Implementation

This chapter informs about different methods used during the implementation work to reach established goals and demonstrate results. This part will not explain deep technical level of each algorithm used, but in a generic form, to give some overall background understanding about each method that was used or researched.

#### 3.1 Application Programming Interface

Most of the time large software solutions cannot offer all the possible features requested by their users, as the individual needs and overall knowledge differ from user to user. For the cases like this, software companies provide an application programming interface (API) to give users opportunities creating own applications to adapt their needs. API lets users to communicate with the software without knowing any details of the implementation [11], often it is provided with well documented description of all the possibilities that API offers called documentation.

As a matter of fact, Tekla Structures provides own API that helps different users to integrate their solutions with the software, as well as, third party companies that offer their services to build application using API for a price. At the Tekla Structures developer centre web page [12], there is a list of main benefits that interface provides to the users:

- Customizing Tekla Structures functionality to fit projects.
- Speeding up daily activities by recording and automating user interface actions.
- Increasing productivity in modelling and drawing creation.
- Turning manual routines into automated actions.
- Integrating Tekla Structures to other software.
- Creating intelligent 3D products and tools for building product manufacturers.

API is a vital part of this research, as it enables gathering of data from the models that are taken into the study. As shown before, connections have a high number of different settings and contain multiple parts that at the same time contain multiple settings and configurations. This way, it would be a much simpler task to use existing API rather than trying to manually collect data using any other method.

### 3.2 Feature Selection

Working with large datasets that contain thousands of different variables and fields it is difficult to understand what relevant data for the case is, and what could be ignored or removed, this process is called feature selection [13]. With a growing interest in developing algorithms and machine learning models to predict something, every time more data is being generated and processed. In these projects, often, each dataset consists of an elevated number of variables and properties called features. Using all of them to generate a prediction model could be time consuming and not accurate. At the same time a person conducting the research might be not specialized in given data context which makes understanding of different issues and finding solutions even more difficult. Because of the feature selection it is possible to reduce context of variables and give a better data context separating “garbage” from “important”. Therefore, the most common benefits of using feature selection are:

- Improving performance and accuracy
- Cost-effectiveness
- Better understanding of the generated data

At the same time, it is important to state that feature selection should be balanced, meaning that removing many variables would make the end model more easy to interpret, making the future prediction accuracy low, at the same time using too many features would make a model difficult to interpret, but accuracy in most of the cases would be higher, as there would be more flexibility.

For different objectives there are different ways using feature selection methods, that's why they can be separated in three different groups:

- Filter methods
- Wrapper methods
- Embedded methods

**Filter Methods** [14] could also be referred as **Single Factor Analysis** are methods that evaluate each variable of the dataset for its relevance towards the result. One of the ways to evaluate is to find the correlation between given variable and the targeted outcome of the operation. At the same time, important note, filter methods do not prevent multicollinearity, a fact that one variable could be an outcome of combining multiple other

variables, which could cause different issues in the future analysis and prediction outcomes. The most commonly used filter methods are:

- **Pearson's Correlation.** A method to measure relation between a single variable and the output measured between -1 as the lowest and 1 as the highest.
- **Linear Discriminant Analysis (LDA)** method finds a linear combination of a variables group that separates output into different classes.
- **ANOVA** is similar to LDA method, but instead it tries to find a linear combination of other features (variables) as one from the given set.
- **Chi-Square** uses frequency distribution of different features to establish a correlation between them and the output.

**Wrapper Methods** work in a different manner from filter. In this case, a group of random or user defined variables are selected to calculate expected result and verify how accurate it was. Based on that, it is possible to select a different set of variables, add or remove existing ones and execute again the same tests from before. In the end, this process becomes a simple search problem of finding the best set of features with the highest accuracy score. The most commonly used wrapper methods are:

- **Forward Selection** is a continuous iteration over the given features, starting with zero in the list. Each iteration adds a new feature that gave the best output result until further selected features do not give any improvement in accuracy.
- **Backward Elimination** is opposite method from Forward Selection. It starts having all the features in the list and eliminates the least significant that improves the accuracy until there is no further improvement observed.
- **Recursive Feature elimination** is a method that tries to find the best combination of features every iteration. It constructs a new model and saves it each evaluation process with the accuracy result. After all the possible combinations are exhausted it stops and gives as a final outcome the highest accuracy combination.

**Embedded Methods** usually combine both filter and wrapper techniques. These methods are inbuilt into learning algorithms and there is no selection or elimination of the variables, there is regularization parameter that assigns low and high weights to each variable. The most commonly used methods are:

- **Lasso Regression** a method that adds a penalty to each irrelevant feature making the weight close to zero, this way they would produce the lowest impact during the prediction of output.

- **Ridge Regression** adds a penalty weight, which equals the square of the magnitude of coefficients. All coefficients are shrunk by the same factor (so none of predictors are eliminated).

Feature Selection might be a difficult job at the beginning, but with numerous methods and solutions being present and more new solutions are being build, there are more ways to understand and prepare large amounts of data for post-processing. One of a very commonly used techniques to generalize different variables into a group or so-called clusters are explained in continuation.

### 3.3 Clustering

Selecting and grouping relevant variables from large datasets is a vital step in data science, for this reason there are multiple methods to do it. One of commonly used methods is clustering [15], in other words, separating values into separate groups based on grouping technique. There are multiple methods of clustering data and all of them have different approach on data that depend on the tasks to be solved. The most popular methods are explained in continuation.

#### 3.3.1 Connectivity-based Clustering

Connectivity-based Clustering [16] is one of many types that allows separating data into groups. It is often called hierarchical clustering due to data grouping is being assigned based on the distances or so-called connections. The main idea behind this method is that objects with less distance are more related than those with higher distance. There are mainly two different approaches to this method. First one, called bottom up, in this case, first, all the values are separated in their own clusters. Next, a distance value is defined by the user that indicates maximum distance between values in the same group. Iteratively distance between all the points is calculated, and if calculated distance is within established one, they would merge into one cluster. The other method, top down, is opposite from the first one explained before. First, all the values are in the same cluster. Next, a distance is defined by the user. After performing distance calculation, points that have higher value than established by the user move to a separate group. After multiple iteration all the values are separated in relative groups. A graphical representation of connectivity-based clustering is showed in continuation.

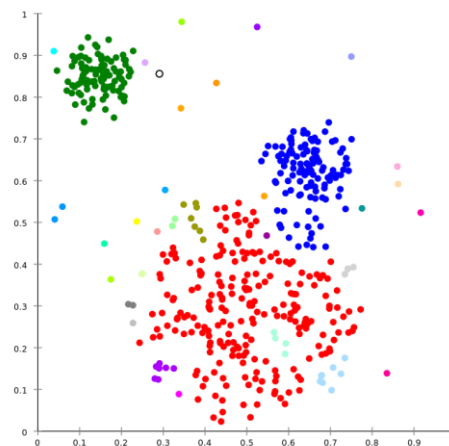


Figure 7. Connectivity-based clustering result.

On the Figure 7 above, a simple graphical representation of a connectivity-based clustering result, showing how data was assigned into different groups represented by different colours. Note that there are multiple single points that formed a separate group, called noise points. The next step having such result would be eliminating noise and distortion points from the dataset.

A great example of graphical representation of a hierarchical clustering is a dendrogram showed on the Figure 8 below. If bottom up approach is used, this diagram can be read that in the beginning there are 25 different values separated into 25 different clusters. After each iteration, if the distances are within limits, they merge into one group until all the variables are separated by exceeding distance value.

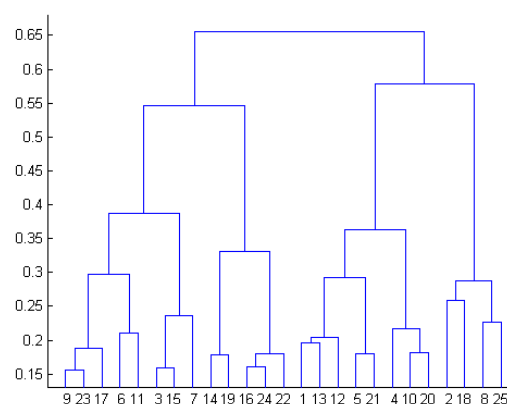


Figure 8. Dendrogram of a connectivity-based clustering.



### 3.3.2 Distribution-based Clustering

Distribution-based clustering [17] is based on statistical distribution model which means a likelihood of all data in a cluster to belong to the same distribution. To simplify the task of distribution there is a popular method to use called Gaussian Mixture Model. Using this method there would be defined a maximum number of groups that is selected randomly and iteratively run classification process. After multiple tries it would give an optimized clustering result that could be investigated and confirmed as an optimal by the user.

Distribution type clustering is a very complex and has a high computing power usage, but it can show correlation and dependency between different variables and attributes. At the same time, for a real data sets there is not a very well-defined model of this method and requires a lot of post work investigation from the researchers.

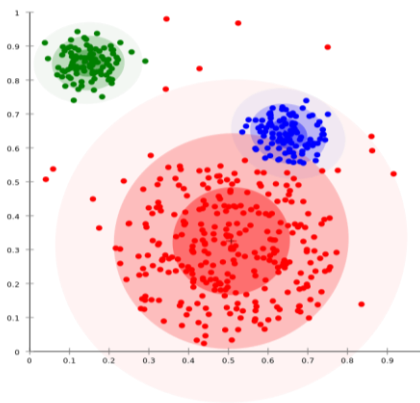


Figure 9. Gaussian distribution-based model result.

Figure 9 shows a graphical representation of distribution-based clustering. It is simple to conclude that two red and blue groups are overlapping each other. This indicates that there is a correlation between data points or a possible dependency between two of them. In cases like this, there are different correlation methods to find what values are related and following that regenerate clusters again.

### 3.3.3 Density-based Clustering

Density-based methods [18] separate data into groups of highest density areas. There are multiple methods that use this technique such as: DBSCAN and Mean-shift. All of methods rely on calculating amount of points around a single point within a certain radius. All these methods have a problem related to establishing borders of the calculated areas.

Each iteration might provide a constant decline in density without any drastic change therefore it is another problem to determine when this area ends. For this reason, there are different performance drawbacks on these methods.

On the following Figure 10 a simple graphical representation of how dense-based clustering result looks.

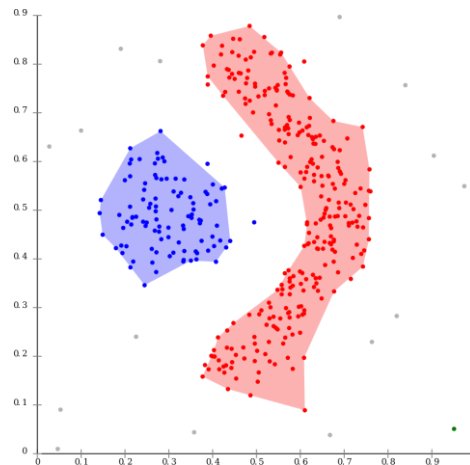


Figure 10. Density-based clustering result.

As it was said, these methods define areas, and it can be seen from the Figure 10. There are two different clusters defined, also some noise points were not counted in any of the groups because this method has its own inbuilt algorithm to eliminate noise points.

#### 3.3.4 Centroid-based Clustering

The last type, explained in this chapter of cluster analysis, is centroid-based [19]. These methods rely on predefined “k” number of clusters given by the user. Using this number, algorithm calculates given number of clusters calculating centroid point that represents centre point of the cluster. These points might not be from the data set. Based on distance from the centre to the given data point, each point is assigned to a cluster. The most common issue in these methods is to find an optimal number “k” for the most optimal results. One of the ways to find optimal number is to iteratively try to use different “k” values until a reasonable or the best result is obtained.

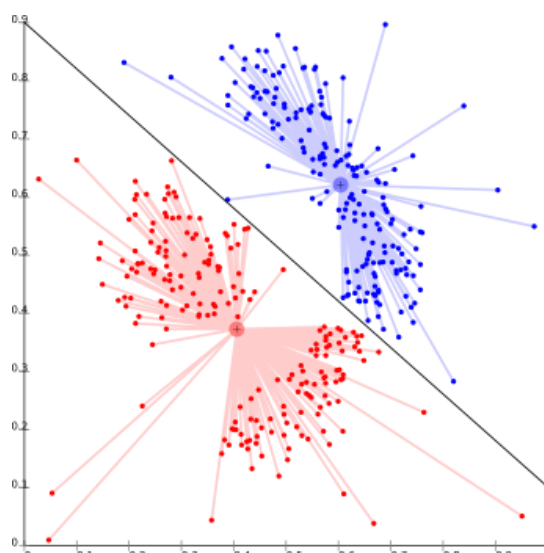


Figure 11. Centroid-based clustering result.

On the Figure 11 as an example, a result of Centroid-based clustering using two centres ( $k=2$ ). Two centre points were calculated from the whole data set and all other points (data represented as a point) were assigned to each cluster based on the distance. This method has issues with nonlinear distribution, despite the showed example, where it is possible to separate both clusters using a line that is drawn on the figure. At the same time, this example could be done using different “ $k$ ” value, it would work well using three or four clusters too.

### 3.4 Similarity Methods

A constant problem in data analysis and data mining is a concept of similarity and what is the most reliable way of measuring it given specific dataset. Similarity [20] defines how object are alike. In a data mining or data analysis fields similarity measure is a distance, where dimensions are the object features. Less distance means the objects are more similar, more distance less similar. Usually similarity is measured in a range from 0 (for lowest or opposite) until 1 (for highest or identical). Note that, measuring similarity or distance should be done using features that are valuable and could give a potential benefit, it is important to study and filter what features to use before assigning dimensions and calculating distances.

There are plenty of different methods to measure distances between objects and groups of objects, but in this chapter, there will be mentioned only five that are relevant to the current research.

First method, which is the most basic and most commonly used in vector space is: Euclidean distance [21]. This method calculates a distance between two points on a straight line, also represented by the mathematical formula at Figure 12.

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^n (q_i - p_i)^2}.$$

Figure 12. Euclidean distance formula.

This formula shows that distance between two points  $p$  and  $q$  is the square root of the difference of their coordinates power two. This method can be used to calculate distance in any number of dimensions, thus making it fit very well with data mining purpose. At the same time, this method has a drawback in performance due to the square root calculation. Possible solution is to use squared Euclidean distance. Removing the square root help removing computing complexity and in terms of finding similarity, if all the other calculations are done using the same method, all the results remain reliable.

Second method is more complex than previous, called Manhattan distance [22], also referenced as Taxicab distance. This method has a different approach and, in some fields, improves early stated Euclidean distance, this method finds a distance between two points as a sum of the absolute differences of their cartesian coordinates. In other words, it's a sum of the difference between each coordinate of the points for example: x, y, z if it is in three dimension space. The generic mathematical formula is given in the following Figure 13.

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^n |p_i - q_i|$$

Figure 13. Manhattan Distance formula.

Represented in a 2D space, shown on the following Figure 14.

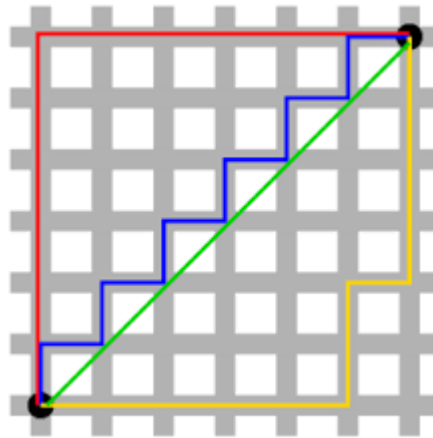


Figure 14. Manhattan distance example.

On Figure 14 red, blue and yellow color lines represent shortest distances between two black points. Green line represents Euclidean Distance, just to show differences between two methods. This approach helps solving distance issues in different areas such as map applications or chess game automation.

Third method is Minkowski distance [23] which generalizes previous discussed Euclidean and Manhattan distances. Mathematical formula for this method is on Figure 15.

$$\sqrt[p]{(x1 - y1)^p + (x2 - y2)^p + \dots + (xN - yN)^p}$$

Figure 15. Minkowski distance formula.

In this formula, the distance is calculated between points X and Y with N number of coordinates in each point and  $p$  is the order of Minkowski Distance. Most often the order values that are used are: 1, 2 or  $\infty$  (infinite). Using  $p = 1$  gives the same results as Manhattan distance, value 2 produces same result as Euclidean Distance and  $\infty$  will give the same as Chebyshev Distance.

Next method is Cosine similarity [24]. Before explaining the method, it is important to mention what is dot product between two vectors. The formula shown below on the Figure 16 represents dot product of two vectors A and B in algebra.

$$\vec{A} \cdot \vec{B} = \sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 \cdots + a_n b_n$$

Figure 16. Dot product of two vectors a and b.

Important to understand from this formula that the result of the dot product is a number and not a vector.

In geometry, more precisely Euclidean space a vector has a magnitude and a direction which usually drawn as an arrow. Based on that, the algebraic formula given before was adapted to fit properties of the Euclidean vectors as showed on following Figure 17.

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \|\mathbf{B}\| \cos \theta$$

Figure 17. Euclidean space dot product formula.

Following before stated, that the result is a number. Dot product between two colinear (angle between vectors is 0 degrees) is 1, and if the angle is 90 degrees the results would be 0 and for an angle 180 degrees is -1.

This formula can be derived to find the cosine of the angle, thus following that, Cosine similarity is represented as following Figure 18.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

Figure 18. Cosine similarity formula.

In other words, similarity is measured based on the angle between two vectors ranging from 0 for the lowest similarity until 1 for the most similar or identical results. For a data analysis and data mining applications, usually datasets contain high number of features and Cosine similarity can be applies to any number of dimensions. At the same time, this method has low complexity, the only rule is that each vector should be non-zero. One often used application of Cosine similarity in data mining is to find cohesion between clusters.

The last similarity method explained in this chapter is Jaccard similarity [25]. This method is used to find similarity between sets or groups of objects, compared to the previously explained ones. Before explaining how this method works, first sets, cardinality, intersection and unions are explained for a general understanding.

Sets, as previously mentioned, are unordered collections of objects where a collection with members A, B and C is noted in brackets {A, B, C} and satisfies that {A, B, C} = {B, A, C} = {C, B, A} and other possible combinations of these three elements. Cardinality is the number of elements in a set and is denoted as  $|A|$  for a set with name A. An intersection between two sets A and B denotes as  $A \cap B$  and the results are the elements that are in both sets. A union between sets A and B denotes as  $A \cup B$  and the result are the elements that are present in A, in B or both A and B.

Returning to the Jaccard similarity. The mathematical formula of which is given in continuation.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Figure 19. Jaccard Similarity mathematical formula.

From the formula and basic concepts explained before, Jaccard similarity between two sets A and B is division between cardinality of the intersection and cardinality of the union. The results are in range of 0 for the lowest similarity and 1 for the highest. Note that for the empty sets A and B assumption is that similarity is highest, meaning that it is 1 but this should be always handled due to a possible division by 0. This method is often used in machine learning applications, for example image identification application where Jaccard similarity is used to measure accuracy of the object detection boxes.

### 3.5 Alternative Methods

This part explains basic knowledge about different alternative methods that were investigated for a potential use during the implementation. They ended up being discarded or postponed due to the scope of this research.

### 3.5.1 Decision Tree

In a real life there are many different applications for trees, but in recent years they became an important topic in machine learning areas especially due to covering regression and classification problems. A decision tree [26] is a flowchart type structure that represents a tree and helps with decision making as a result, that is where the name comes from. In a decision tree a node represents a check, or a test of an attribute and the branch is an outcome of the test that leads to next node or to a leaf that represents the outcome of a classification task. The following Figure 20 shows different elements of a decision tree and its graphical representation.

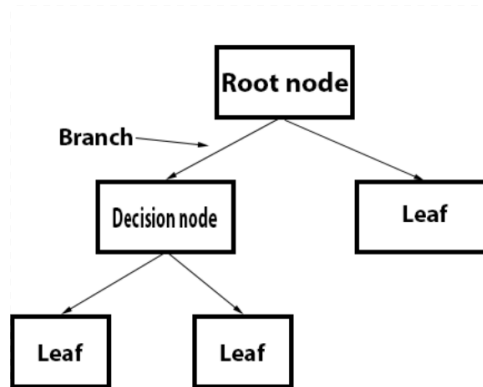


Figure 20. Basic representation of decision tree and its core elements.

When talking about decisions trees there are different terms that are used:

- Root node: represents a start point of decision-making process that contains all the samples.
- Decision node: is a result of splitting a node into multiple nodes that contain a test or evaluation of the given information.
- Branch: is a subsection of the tree that guides to the next decision node or a leaf.
- Leaf: this is the last node in the tree that usually represents the end of decision tree and contains the result of classification.
- Splitting: is a process of splitting a node into sub nodes connected by branches.
- Pruning is a process of removing sub-nodes of the tree, opposite to branching.



During generation or creation process of the decision tree there are different methods used but one of the most used is called recursive binary splitting. During this method, all the given features are considered and for each of them cost function is calculated. Cost function is an algorithm that evaluates how important is feature towards classification. In the end, the best cost features are selected to generate a decision tree. This method is also called greedy due to its nature of selecting always the best cost. Usually, cost functions used for classification and regression find the most similar groups and minimize the error of classification by using squared difference.

In machine learning, data sets are used to have high number of features. Using decision trees might take a vast amount of time. There is a big change that generated tree would simply learn a pattern to classify given data with the expected result. To prevent this from happening there are methods as decreasing minimum amount of cases to generate a decision node or set a maximum depth of the tree which will also help with performance due to a shorter time traversing the tree.

In cases when performance start to decline, pruning is a good solution. Pruning, as described before, is a process of removing decision nodes that evaluate features with least impact on the classification. This could be done starting from the root or in reverse, from the leaves, but most common practice is to start from the leaves and up the root which is called: reduced error pruning.

To summarize decision trees, there is a list of benefits that they give:

- They are easy to understand.
- Rules are easy to generate.
- Decision trees perform inbuilt feature selection.
- Require little effort in data preparation.

At the same time there are different disadvantages using decision trees as:

- They can learn classification if there is no control over depth, producing low accuracy results.
- Any data variation could lead to a different tree generation.
- Greedy algorithm is not always the best cost function as it is not always the optimal solution.

- Calculations are more complex if there are large number of classes.

### 3.5.2 Neural Networks

In computer science many different methods and algorithms were derived from the real-world processes. Neural networks [27] are not an exception. This technique can be heard everywhere nowadays, everyone is interested in this topic, but the concept was first opened in late 1943 by Warren McCulloch and Walter Pitts by creating a first computational model for neural network.

The name of these networks comes from the biological structure of human brain. Like in human brain, there are millions of neurons interconnected and passing information from one to another until reaching decision and giving feedback to different human body organs. In artificial intelligence, neural networks consist of multiple neurons interconnected and information flows always forward. Each neuron contains a mathematical function that classifies input information based on the architecture and the result is passed forward reaching the last neuron in the structure.

During neural network model generation there are two types of data sets: one contains training information and other contains testing information. The model trains using training set where each neuron will run specified architecture. This training process is usually iterative process meaning that training set will be used multiple times until the testing process minimized error between testing set results and predicted output results given by the network. This type of learning is called unsupervised which means that there are no pre-existing labels on the data and the neurons will compute them. Since the method was found more than 70 years ago it got very popular only recently and found its use in multiple areas and applications such as: image recognition, text recognition, transforming text to speech and many more.

Neural networks are very popular in machine learning and building artificial intelligence. This fame gave a start to a wide variety of new areas in software engineering, as well as, provide multiple sources of information to everyone. Although, they seem as a good approach to solve different tasks, but before using the neural networks it is important to understand different drawbacks that this method contains:

- Building an accurate model requires a lot of training data. Often to have a very well adapted model, training set should contain as many variations of data as possible.

- Running neural network model generation in excessive amount of iteration might produce overfitting. Overfitting is a result of over training model and as a result it maps every input data to output.
- Neural networks do not provide any visible explanation how the classification was done, there is no practical way to find how and why a given neuron or the whole network produced the output.
- Requires a large amount of computing power to generate a network model. Growing amount of data and extensive neural network architecture that contains hundreds or thousands of neurons requires a lot of processing resources.

### 3.6 Project Plan

In the end, project plan should be established before starting the implementation part. The established plan is showed on a Figure 21.

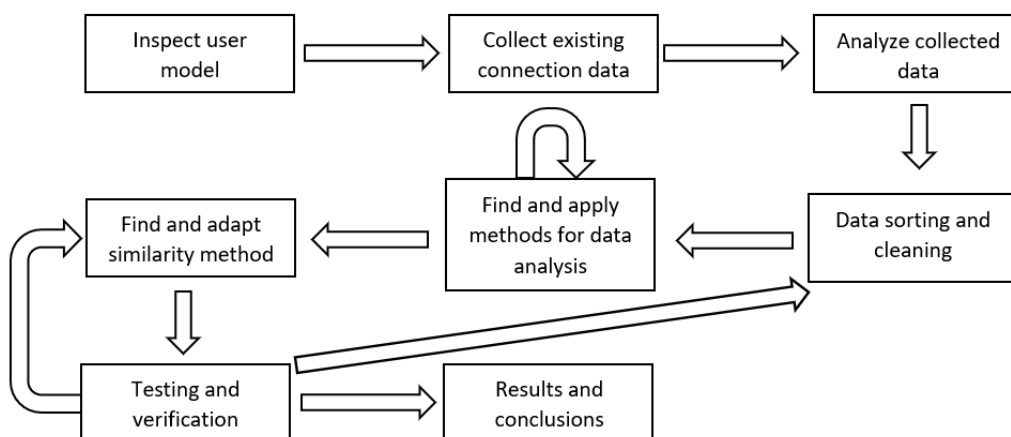


Figure 21. Project plan.

In this research project, to find out how to find similarity, a finished user model should be analysed first, to know what objects are connected, how many connections, is there any autoconnections set up in the project and other details to get familiar with.

Following that, existing connection data should be gathered and analysed just by human eye, checked if there is anything visible that should be excluded or separated from this information or there is a clear repetition of some attributes or settings. After collection and analysis, it is possible to search for different existing methods that would analyse collected data in depth. This would provide a simplified view of the collected data set. At the same time, it would be not optimal to use only one method, in this case, results might

be not clear, or data is not fit for the selected method. In case when selected method does not fit project needs, other method should be applied.

When data was successfully analysed for the first time, it would be possible to find out what is the most relevant information and what is not relevant at all, so the raw data could be sorted, cleaned, etc. to avoid different data issues in future. With a cleaner data, it would be the time to apply for second time different analysis methods used before and investigate changes or improvements.

Next step is to find out what similarity methods could be implemented for this data, what existing popular methods there are and how to apply them on existing data. After selecting a method, it needs to be adapted to collected data and desired output. Each method should be checked as well and if it does not fit, another should be selected.

In the final steps, having implemented similarity method, it is the final step to test and verify how it is working with a user model. The test is carried on basis of comparison of what is expected (the existing connection on the model) and what was created, using selected similarity method. This comparison should check what type of connection was created, what setting were applied and is the result visually correct from a detailer points of view. It is important to mention, that often there are different possibilities of valid connections between set of construction objects. It is a job of a professional detailer to decide what would be the best fitting option in the given situation and if that is optimal. After running tests, it might show that the selected similarity method did not work as expected, so there is also a need to find and use another method that could provide with better test results. At the same time, it is not excluded that it might be needed to return to data analysis or sorting and cleaning step to reorganize and rework collected model data and after that continue with similarity methods.

Finished testing and verification would provide with an evaluation of the implementation. It would make it possible to arrive with conclusions of the research and document results, finding and future work improvements to be done.

## 4 Implementation

This chapter explains about how the implementation part of the research was done to accomplish established goals within the given scope. The chapter follows given project plan given in previous chapter. It is important to note that due to some company privacy policies detailed method names, how they were used or developed are not disclosed.

### 4.1 Case Study Model

The first and one of the most important steps is to understand given 3D model and what kind of information and connections are present. Because of this, case study model was opened using Tekla Structures software to visualize data to be worked with. The most important objects from this information are connections. Easiest way to know how many objects of a specific type present in a model is to use specific option in filtered selection. Setting up the filter selection to select only components in the model is possible with keyboard key combination CTRL + A to select all the components in the model. The result of this selection is given on the Figure 22.

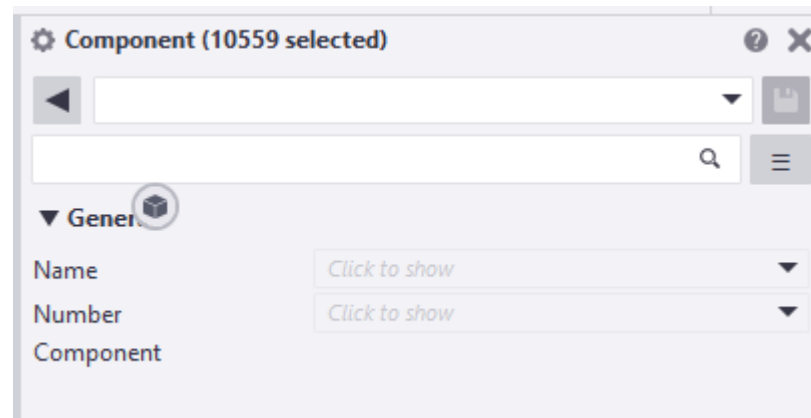


Figure 22. Number of components in the case study model.

In this model there are 10559 components in total. Components are not only connections, but this also includes detailing and custom components that users create themselves to ease repetitive work of creating different parametric objects. In the figure above, there are also two labels Name and Number. In case when selected components have the same name this field will show the value of the name, same goes for the number value field. When there is no number or name it means that there is no unique value for

name field and/or number, this way it is possible to guess that selection contains multiple combinations of different components.

Investigating different components in the model it is possible to zoom in and check what objects are inside the components and where they are placed to have an overall view on how they are used in the given context. During the model inspection phase, it was found out that in the model many of the possible connections were exploded. Exploded connection is a functionality in Tekla Structures when a connection can be removed but the objects inside it will remain in the model as standalone parts and could be modified individually. This observation brings up many different thoughts that given configurations and connection catalogue might not be enough for the users. This brings up possibilities for future research in this topic.

To investigate further with what number of connections used in the model Tekla Structures Open API was used. This interface offers multiple possibilities to filter different type of objects on a deeper level than user interface of the software. For this reason, a small application using C# programming language was created to query all the object of connection type. Figure 23, below, shows the output result.

```

Number: 14----- Repeats: 60 times and Name: Joining plates
Number: 144----- Repeats: 522 times and Name: End plate
Number: 90----- Repeats: 52 times and Name: Railing plane to plane
Number: 89----- Repeats: 678 times and Name: Railing beam to plane
Number: 22----- Repeats: 1395 times and Name: Tube crossing
Number: 20----- Repeats: 659 times and Name: Tube gusset
Number: 184----- Repeats: 1119 times and Name: Full depth
Number: 27----- Repeats: 133 times and Name: Stiffened end plate
Number: 41----- Repeats: 155 times and Name: Cranked beam
Number: 146----- Repeats: 342 times and Name: Shear plate simple
Number: 131----- Repeats: 169 times and Name: Column with shear plate
Number: 30----- Repeats: 1136 times and Name: Seating

```

Figure 23. A list of different connections present in the case study model.

Each row starts with the number of the connection from the inbuild catalogue, then number of times it repeats in given model and the name. The total number of used connections in the model is 6420 which is lower compared to what was showed on the Figure 22. The difference means, like it was explained, that components are not only connections but also details and custom components and other objects. At the same time, there are only twelve different type of connections being used with multiple repetitions.

Due to high number of repetitions of the same connection type, more in-depth analysis was made. Extending existing application created before, gather every object that forms each connection. The result for a specific connection number 144 is shown in the Table 1 below.

Table 1. Number of repetitions of the same objects present in connection number 144.

Repeats times	Contains objects
183	Fitting, ControlPoint, ControlPoint, Beam, Weld, Weld, Weld, ControlPoint, ControlPoint, BoltArray
139	Fitting, ControlPoint, ControlPoint, Beam, Weld, ControlPoint, ControlPoint, BoltArray
179	Fitting, ControlPoint, ControlPoint, Beam, Weld, Weld, ControlPoint, ControlPoint, BoltArray
4	Fitting, BooleanPart, ControlPoint, ControlPoint, ControlPoint, ControlPoint, Beam, Weld, ControlPoint, ControlPoint, BoltArray, Weld
9	Fitting, ControlPoint, ControlPoint, Beam, PolygonWeld, ControlPoint, ControlPoint, BoltArray
5	Fitting, BooleanPart, ControlPoint, ControlPoint, BooleanPart, ControlPoint, ControlPoint, ControlPoint, ControlPoint, Beam, Weld, ControlPoint, ControlPoint, BoltArray, BooleanPart, ControlPoint, ControlPoint, BooleanPart, ControlPoint, ControlPoint
1	Fitting, ControlPoint, ControlPoint, Beam, Weld, ControlPoint, ControlPoint, BoltArray, BooleanPart, ControlPoint, ControlPoint, BooleanPart, ControlPoint, ControlPoint

From the results given in the table it is possible to conclude that there are many variations inside the same component due to user customization. There are 522 connections with

number 144, but there are seven different cases in what objects they contain which gives a lot of variations. Although, there are many objects there is a visible repetition of same object types as: beam, weld, fitting that will be beneficial in the next steps.

From analysing given model there were multiple observations made and the next step is to find the best methods for further data analysis and grouping of the data to fit into similarity methods to obtain the best results.

Next step towards the goal is to analyse what kind of features all connections share and what are unique or irrelevant that could harm results in the future. Each connection has about 700 attributes but not all of them are used, this is decided on the connection implementation algorithm, this will not be discussed in this research. Every part object has own attributes that define its geometry and basic connection creation workflow uses those parts to create a connection. From this thought, it is possible to conclude that main and secondary parts are the most important pieces at the connection design process.

Data cleaning and enhancing should contain only relevant information. Some time was spent on analysing different attribute lists that parts and connections consist of. There were several different wrapper feature selection methods used to obtain the most valuable properties and well as least ones using C# language. A list of different attributes that should be omitted was made in the end. The list is extensive and only the most important excluded features will be discussed.

First important group of attributes that were ruled out are company and user information. Some of the attributes contain company specific settings and attributes as well as high level user settings that should be removed before taking the data further. This type of features is irrelevant in deciding connection design and, moreover, this is a private information that belongs to the company and users. The other important set of attributes is global location in the model. Many different properties include exact position of the object in the model that could harm future accuracy and decision-making process. It should be considered irrelevant information of exact point in space. There should be no dependency where each element was located, but for example, location information between related objects would be a very beneficial to establish position relations between objects.



Important features that were considered are geometric properties such as: height, width, form and shape. This information is considered one of the most important factors to decide on what type of connection should be created based on the shape of the neighbouring parts. The other is the structure of the connection and which is main and secondary parts.

The next step, as it was discussed in project plan, was to find methods to analyse and adapt data before feeding it to selected similarity method. Due to high number of floating-point numeric attributes and properties it would be even more complicated task to evaluate similarity. Analysis showed that different values should be grouped by their proximity. In this case, the best way to proceed was to use clustering techniques. For each feature from the list there were different clusters created, assigning each numerical value to a group (cluster). To select the final clustering method to use in the final prototype different connectivity-based and centroid based options were implemented and the one with most promising results was selected.

Having all the data cleaned and sorted, the next step was to implement and select a similarity method to evaluate the similarity between connections. In the second chapter, current state analysis, there was a short description of an ongoing research about auto-completing a set of construction objects. Due to limited resources and the scope this was a vital step to integrate existing prototype solution with the current research.

After a successful integration of existing prototype, it was possible to generate an output due to the existing similarity comparison between each construction object (part) in each connection. The results of prediction were not accurate and, as expected, more work had to be done. To provide a more accurate results it was decided to evaluate not only similarity between objects inside the connection but also find out a relation of their respective locations to each other. In other words, taking a main part of a connection and find out how the secondary parts and positioned around it. This was implemented in the same form as previously. First, collecting information about positioning, then creating clusters for better accuracy and tolerances and as a final step feeding this information to the main program.

Adding more functionality gave results towards the output accuracy. But, it was failing at the times during the creation of connection because of wrong selection of the main part. This gave many errors showing newly created connection in a wrong place or orientation

in the model. This could not be accepted as an outcome. Next step was to add a new processing algorithm that find different combinations of all the selected parts and finds out which selected object is the best candidate to be the main part. After that, continue with already implemented similarity evaluation process. This gave a very promising improvement in accuracy and correctness of a new created connection but at the same time increased processing times to find possible matches. To give an example, before improvement the time spent on finding similar match was under 1 second and after, over 30 seconds. So far issues with computing times were not in the scope and no investigation or implementation will not be pursued further.

This chapter did not mention anything about the computing times and performance issues yet. There are a lot of issues connected to the time spent collecting existing data due to a very high number of attributes that each individual construction object has. It was shown that each object contains around 20000 different features and cleaning process decreases them to 19000 which does not reduce complexity. Due to those findings time consumed to process a finished user model could take more than a day. Performance is currently an open question and is not solved in this scope.

In the end, the prototype was ready and different evaluations and verifications were done to prove that it worked. Some of the results are showed in the following Chapter 5 called Results and Analysis.

#### 4.2 Alternative Methods Approach

During the research of different available methods of classification and prediction of data there were attempts of adapting other techniques during implementation process. First approach was done using decision trees. It was explained previously about different advantages and issues that decision trees could provide. Considering all the risks it was given a try to predict the connection using collected data. Python programming language was used for this as it provides many libraries to generate decision trees and was the fastest approach to work with.

The implementation consisted of generating exactly a hundred decision trees to classify connection number (connection type) as the result of classification. For each generated tree, accuracy of the testing and verification was stored. These results were used to pick the highest accuracy tree as a final solution. The best generated tree gave accuracy on the testing samples 92.2% which looked as a promising approach. Continuing further,

this tree could only classify given data to a class that was the connection number (connection type) to create and nothing more. Only the number does not provide all the necessary information to create a connection. To proceed with this approach, it would require generating as many trees as features, which is about 700. This was not something that could be done within the time limit and the scope and was decided to be discontinued at that point.

The other method that was considered because of its current popularity is neural network. This approach gives no explanation on how different rules are generated and how it derives to the result. This would produce difficulties analysing why the results are produced and spend more time on finding out how to improve them. Method like this requires millions of samples to have a good learning ability and produce more accurate results. From the Figure 23, it is clear that the amount of data is insufficient. It is very problematic to get a finished user model because they are private or not in the public domain yet. Moreover, construction companies would have to give permission to use them for the current purpose that would pose more time delays. Due to multiple drawbacks of this method and the project needs it was decided to discard this method from further implementation.

## 5 Results and Analysis

For a better understanding of results, different cases of produced outcomes of the prototype will be shown. Each case contains initial situation, expected outcome and produced outcome by the solution. Initial situation shows what is the selected set of objects given as input. Expected outcome is the target to be created by the prototype. The produced outcome is the result of running prototype using given set of construction objects. As analysis part in this chapter a comparison between expected and given results will be explained in terms of correctness in each modelling situation, e.g. if the produced solution could be accepted as final. Every case that differs will be discussed with an experienced connection design engineer to verify that produced result could be accepted as a valid outcome or discarded [28].

For this task a model that was not used during development process was selected. Given model contains different connection with different number of parts. The overall number of connections and the number of construction objects used to create them is given in following table.

Table 2. Number of connections and objects used to create those connections

Number of connections in model	Number of objects used to create connection
387	2
6	3
5	4

From the given Table 2 it is possible to see that there are numerous connections created using 2 parts, but others are very rare in the given model. This will have a major influence during the result outcomes showed in continuation.

### 5.1 Case 1

The first selected case was tested between a column and a beam as input objects showed on the following Figure 26.

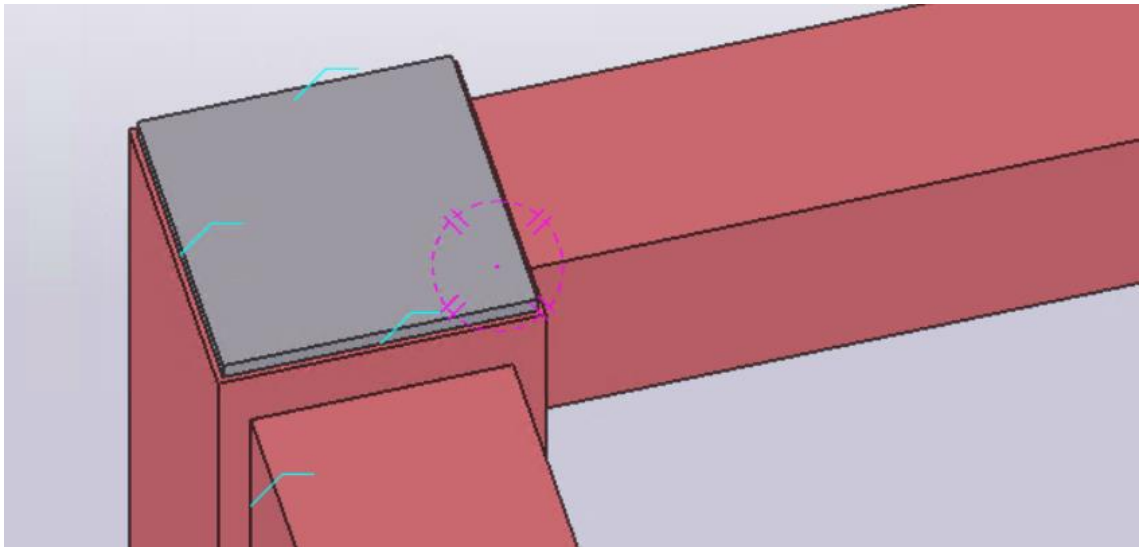


Figure 24. Case 1 initial situation.

In this situation there are some detailing done beforehand on the column that consists of a steel plate welded on top of the column. The interest of connection is between column and horizontal beam on the right side. Solution developed by the owner of the model shown on the Figure 27 next.

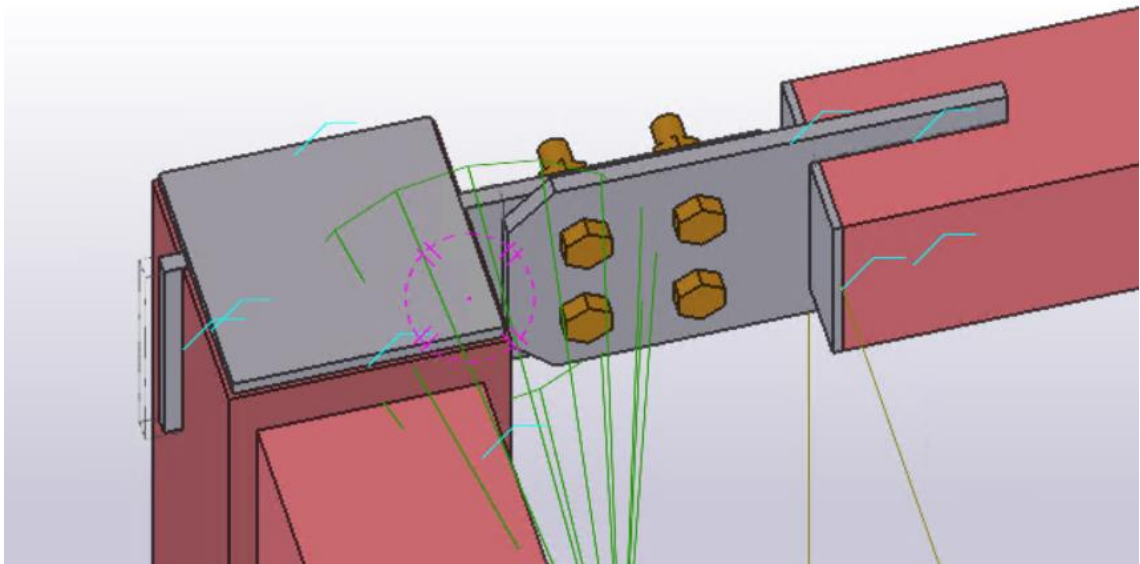


Figure 25. Case 1 expected output.

The expected result creates multiple cuts and two new steel plates that are bolted. Horizontal beam becomes cut to fit the new creates details. There are multiple new welds between newly created steel plates and input parts. This represents a complex structure

to model and adjust to the measures. Following these observations, created prototype was executed using the selected parts and the following result was produced.

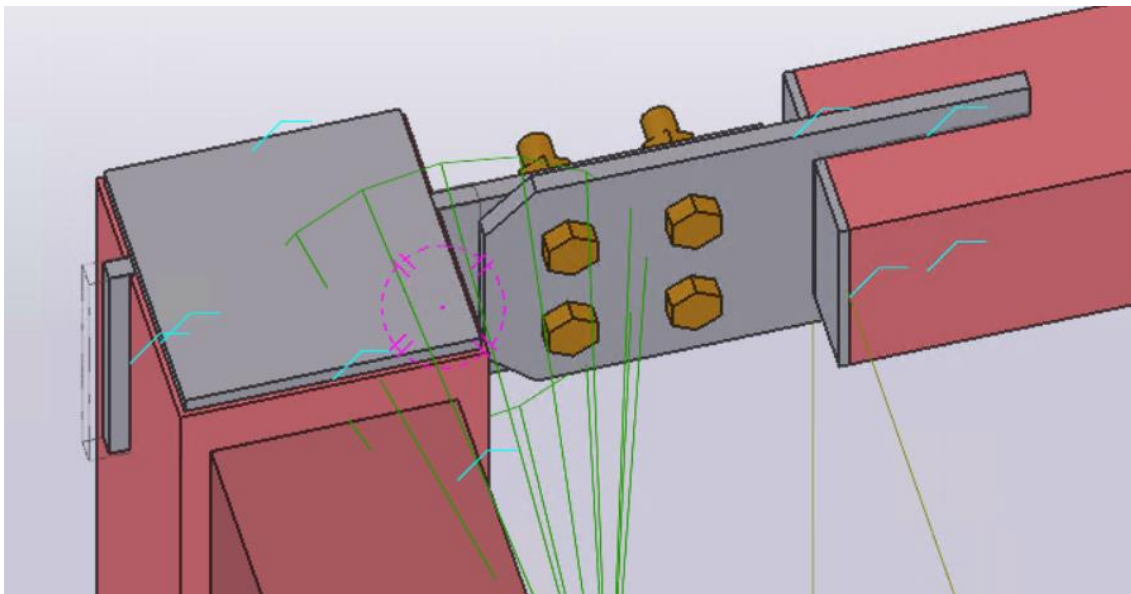


Figure 26. Case 1 produced output.

In the end, even this complex structure was created by developed prototype with 100% accuracy. Accuracy in this case means that the modelled result using similarity was exact copy from the expected result. Just from visual investigation it is possible to conclude that this test case was a success. On the attributes and settings level, all of them were correct and exact as a professional connection designer would set up.

## 5.2 Case 2

Second case initial situation is showed on Figure 29. In this case, input objects are of the same type: beam and column. Both objects have similar attributes and configurations. Despite the first case, selected beam is not located next to one of the ends of the column but in the middle. This case also has different beam direction compared with the column.

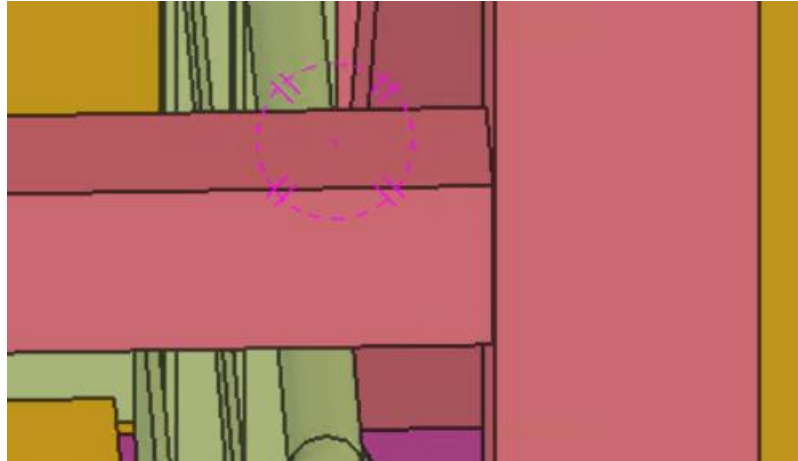


Figure 27. Case 2 initial situation.

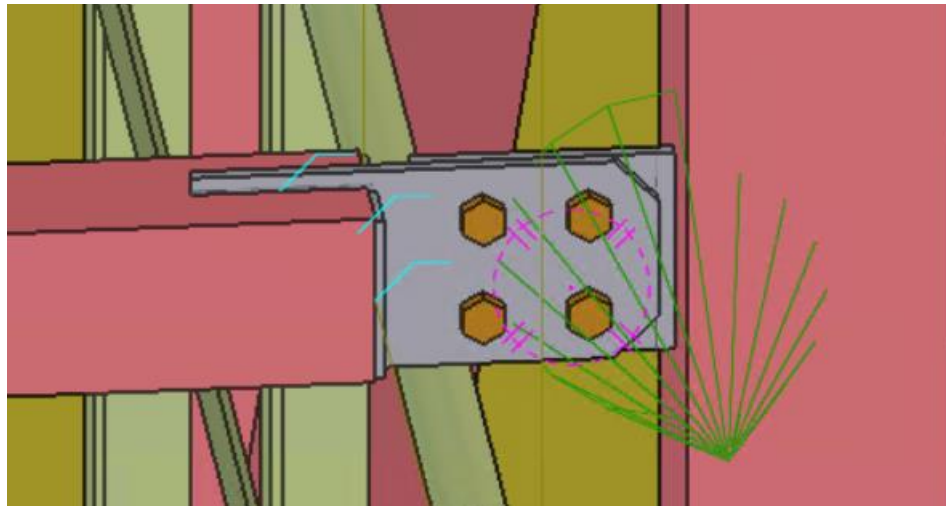


Figure 28. Case 2 expected output.

The following Figure 30 shows what is expected connection outcome designed by a professional connection designer. This case is similar to the first one, but there are differences in number of welds and cuts created between selected objects. The following figure 31 shows the outcome of the prototype.



Figure 29. Case 2 produced output.

From the visual comparison of two figures it is possible to detect differences. The outcome of the prototype solution created a connection with a different direction. Inspection of the created details did not reveal any other differences between two solutions. In this case opinion of a professional engineer was required to verify if produced solution by the prototype could be accepted in a construction project. From the words of the engineer it was established that the output would be accepted as a valid outcome. All the settings of the created connection were correct and there were no design issues detected. This case concludes results for the connections using two input parts.

### 5.3 Case 3

In this test case a set of 4 input objects were used to create a more complex connection. From the before presented Table 2 there were only 5 connections created using 4 construction objects as input. It makes this a very interesting case to investigate how the prototype performs with a low number of similar examples.

Initial input content with 4 objects is shown on Figure 32. This case has 3 similar beams intersecting with a column.



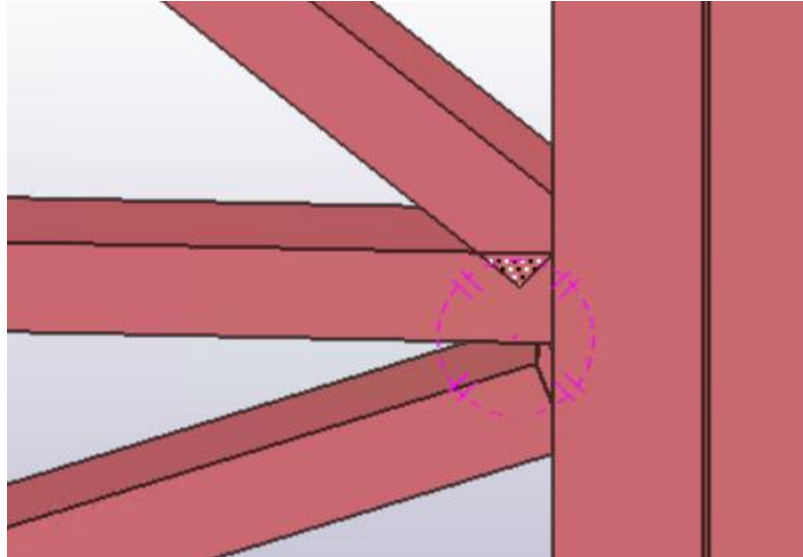


Figure 30. Case 3 initial situation.

In the expected result shown below on Figure 33 all 3 beams are cut and a large steel plate is created in between all the beams and column. Each beam is welded to a smaller steel plate that at the same time each of the plates are bolted to a large one.

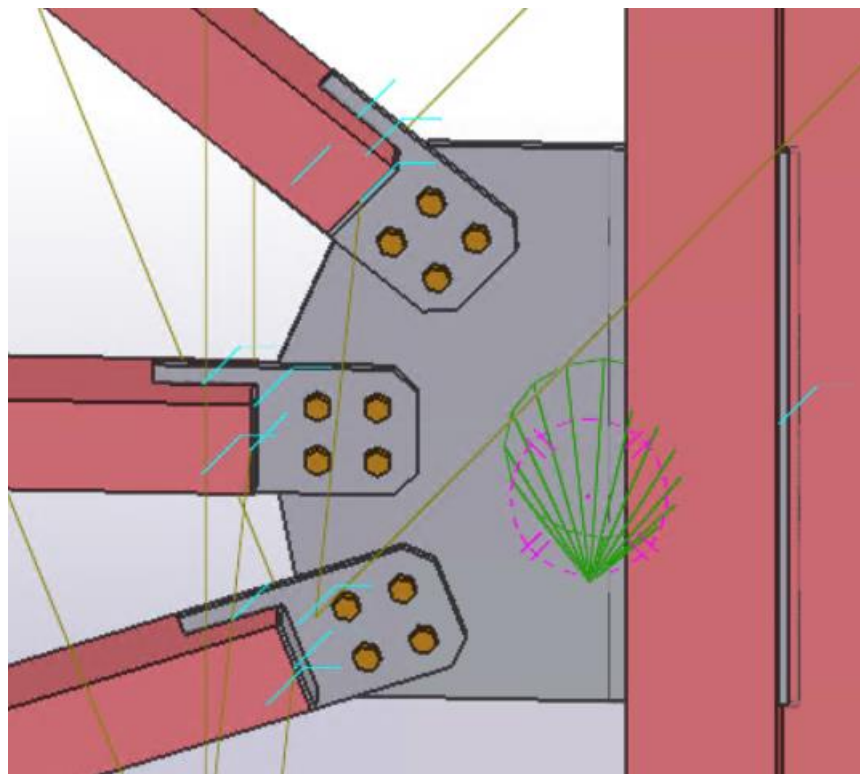


Figure 31. Case 3 expected output.

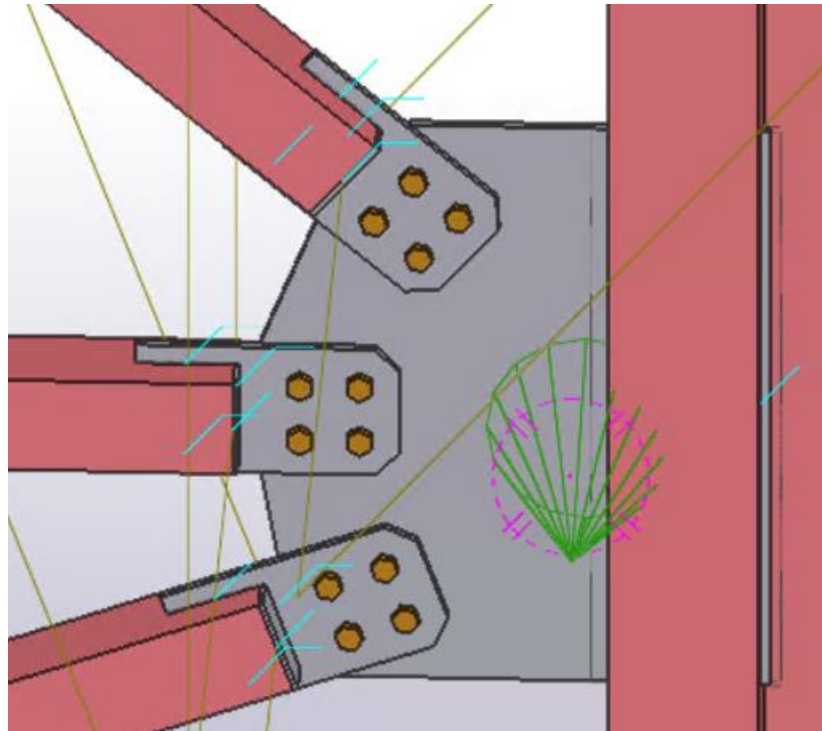


Figure 32. Case 3 produced output.

Running developed solution reproduces a connection shown on the Figure 34. From visual analysis there are no differences. All the details are in correct places, as well as, direction of the connection. Visually there are no issues, comparing applied settings and attributes revealed no differences as well. In this given test case created prototype produced identical result as a engineer would do even having not a large list of similar examples.

#### 5.4 Case 4

Following case input parts are similar that were used in Case 3. There are 4 different objects to be connected. Similar 3 beams and 1 column as was given before. What is different that the parts are oriented in different direction and due to low number of other examples this might produce a different result from previously showed.

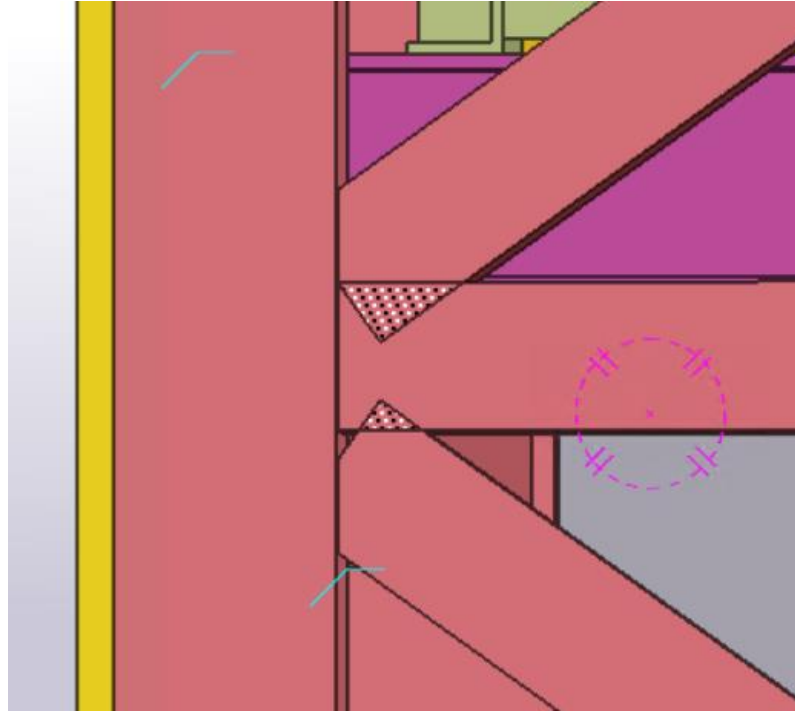


Figure 33. Case 4 initial situation.

In the following Figure 36 expected connection design result shown. This result is very similar to the Case 3 only direction of the connection was changed. There is a large steel plate that is welded to the column and all the beams are cut and welded with smaller sized steel plates which are bolted to the larger plate.

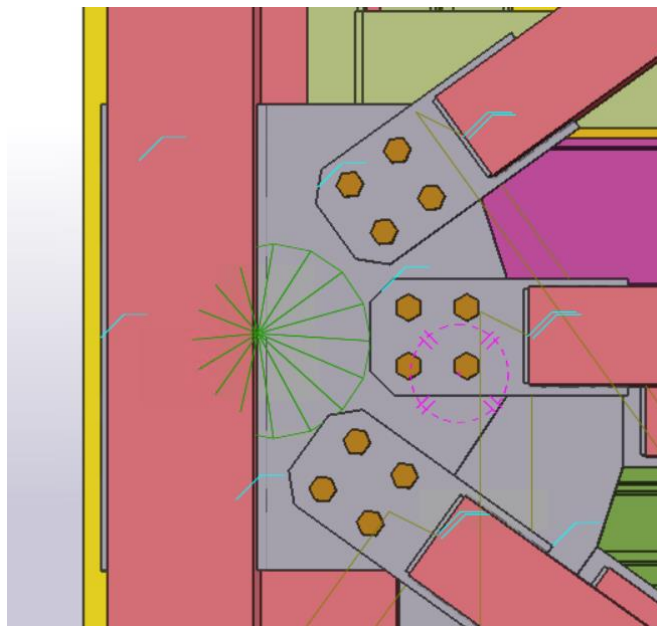


Figure 34. Case 4 expected output.

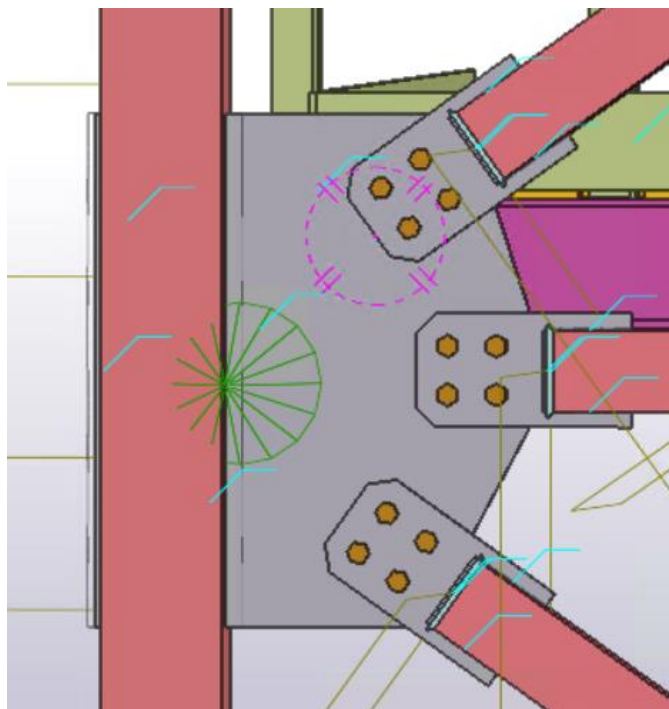


Figure 35. Case 4 produced output.

Figure 37 contains produced result by the prototype. At the first sight it looks very similar to the expected outcome. The problem of this solution is in the large plate that connects beams and column. In the solution given by the prototype the larger plate has bigger dimension that what would be expected. For more clarification on the given solution help of professional engineer was asked.

Opinion given by the engineer concluded that this suggested solution could be accepted if the construction project is not strict on material usage. Most of the construction projects have policies towards minimizing used materials during the construction which benefits the overall budget of the project. Said that, this could be a valid solution but not the optimal for a real project. In the end, suggested solution could be modified using connection settings and due to a correct set up of connection type and different attributes it only required changing few parameters to obtain the same result as was shown of Figure 36.

## 5.5 Case 5

The last case discussed in this chapter is a check if connection between 3 objects would work. This was not tried yet and the number of examples in the given model is very low.

In this case there are 2 beams and a column. This is similar situation to one of the previous tested cases as the column has a welded steel plate on top.

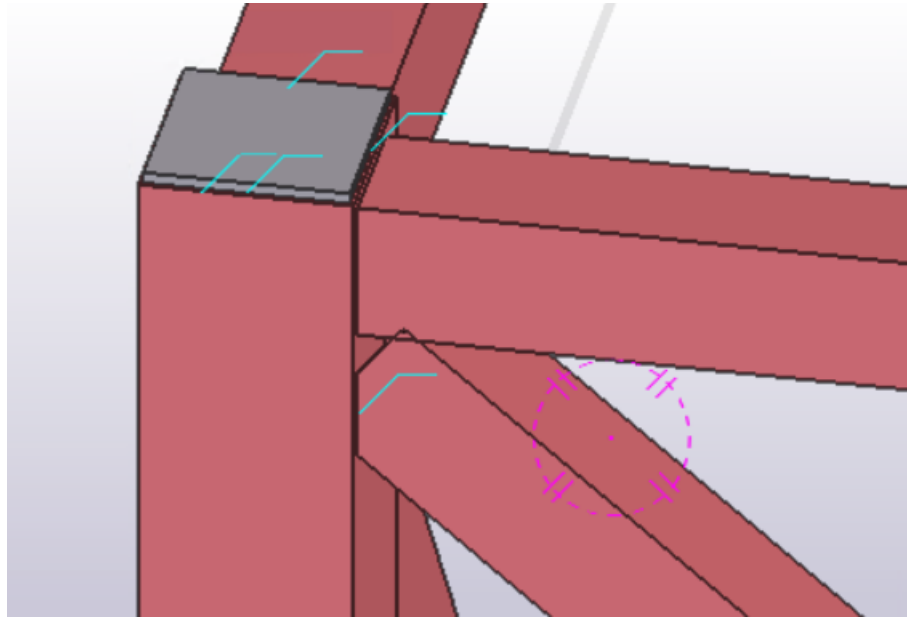


Figure 36. Case 5 initial situation.

The expected result of designed connection would be as showed on Figure 39. The same situation where a larger plate is welded to a column, beams are cut, smaller plates are welded to the beams and smaller plates are bolted with a larger one.

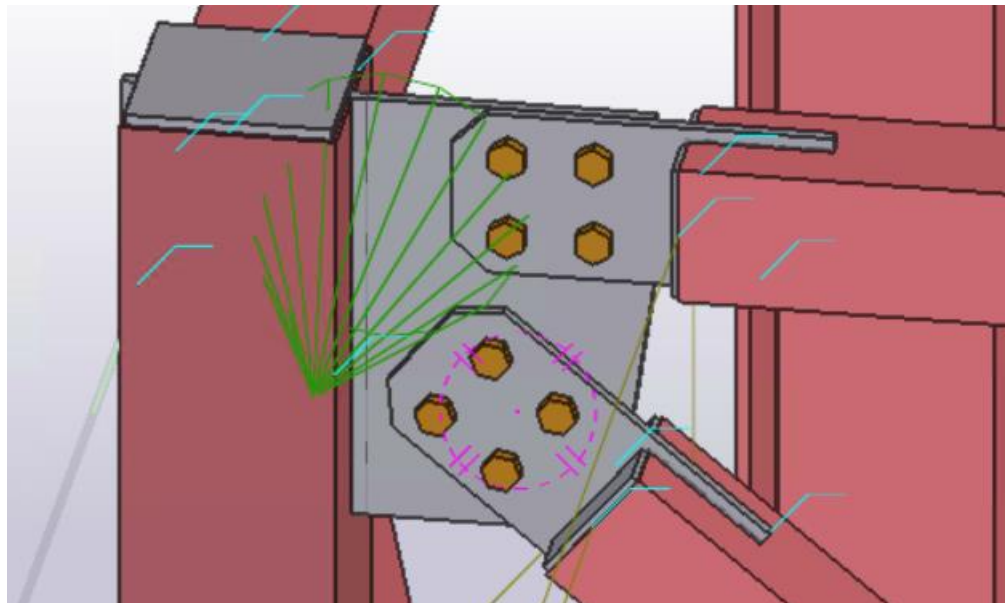


Figure 37. Case 5 expected output.

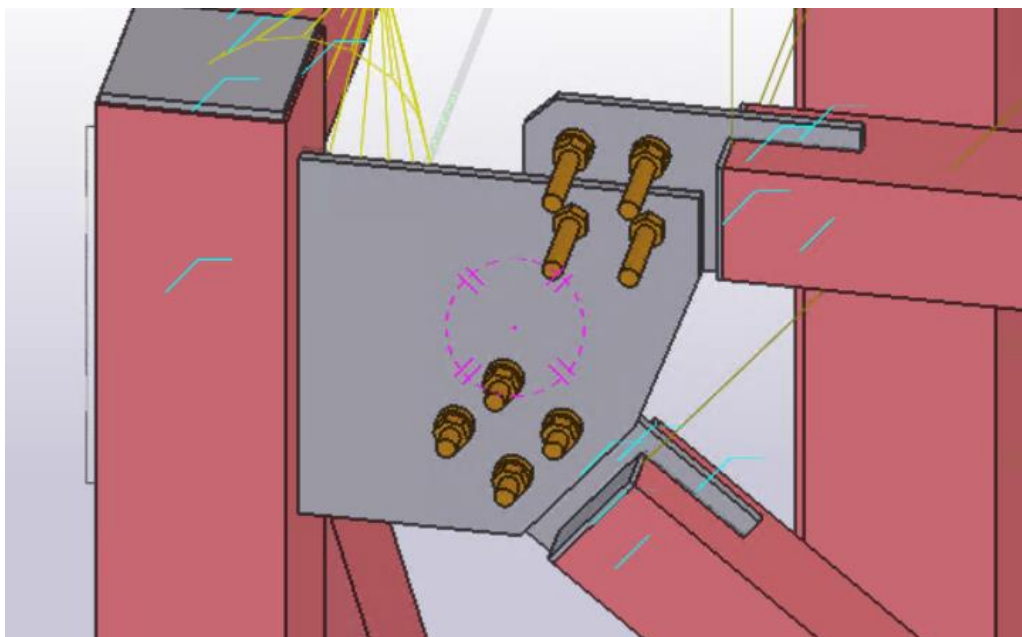


Figure 38. Case 5 produced output.

Figure 40 shows developed prototype suggestion. In this case problems are clearly visible. Plates are not positioned correctly, larger plate is small, bolts do not have correct length and direction is opposite. This is an example of an almost complete failure of this project solution. To get more understanding if this solution gives a possible valid result or not, opinion of connection engineer was asked.

As it was expected, engineer concluded that this solution is not correct and would never pass any construction checks. Due to wrong bolt sizes, size of larger plate and connection between plates overall this connection would fail in a real construction under the weight of steel objects. The only positive conclusion is that the suggested connection had a correct connection number (type). With this information it would simplify the task of finding the correct connection type to use in this case.

Summarizing results and analysis chapter, the outcomes of the prototype were quite promising but at the same time some of the cases were not as good as expected. In the next chapter conclusions and future improvement suggestions are discussed.

## 6 Conclusions

This chapter explains if the research questions established in the beginning of this thesis work were reached by the implemented solution and further work and improvements possibilities are discussed.

One of the main reasons to pursue with this research topic was to increase personal experience with connection design process. It was difficult to understand what exactly is needed in different cases and what are all those available parameters and settings that each connection has. Due to imperfect existing solutions and currently ongoing research project at the company that is well connected with this one, it was decided to give a try and find out a better way to generate connections between construction objects. At the same time, many machine learning techniques provide very promising solutions. This was the other reason of selecting current research topic, to experiment with different learning techniques, data analysis and data mining using large software solution as Tekla Structures.

Following with conclusions, it is important to remember what research questions were established for the given research. The first question was asking if it is possible to find similarities between connections in the model.

During the research, one of the starting points was to investigate the model and existing connections in it. During the investigation of the given construction objects, detailing and attributes that each connection has this question could be answered. Without any doubt it is possible to measure similarity between connections and furthermore find a coefficient of similarity between them.

The second question followed the result from the first one if it gave positive answer. Is it possible to create new connections based on the previous modelling projects?

In order to answer this second question, the prototype was developed. Using the implemented solution described in Chapter 4 (Implementation) it was possible to get the results shown in chapter 5 (Results and Analysis) that can answer the given question. More precisely, the answer to the second question is yes, it is possible to create new connections from the previously made projects. During multiple testing cases some of the results



showed promising outcomes, but some were not. It was seen that amount of data samples are crucial during this kind of work. In the model, used for results and analysis chapter, there were plenty of connections created using 2 objects, but only few using 3 or 4. This made the results much better in case of 2 objects and worse for 3. On a general level this research was a first attempt to answer the two important questions that would enable future development of a more complex and precise solution.

At the end, this research answered both established research questions and showed promising results concluding in a successful prototype. Different potential future work suggestions to be done are listed below:

- Improve time spent on model data retrieval.
- Improve time spent on finding the most similar match between selected objects.
- Try different classification and similarity methods from those that were used in the prototype.
- Create a metric that could evaluate accuracy of the prototype in each model.
- Collect more available user models to be able to use more advanced machine learning algorithms.



## References

- 1 Tegel W., Elburg R., Hakelberg D., Stäuble H., Büntgen U., Early Neolithic Water Wells Reveal the World's Oldest Wood Architecture. 2012.
- 2 Encyclopaedia Britannica. Civil engineering. Internet Article <<https://www.britannica.com/technology/civil-engineering>>. Accessed 20 September 2019.
- 3 Narayan K.L., Computer Aided Design and Manufacturing. New Delhi: Prentice Hall of India. 2008 pp. 3.
- 4 Constructible Trimble, Building Information Modelling. Internet Article <<https://constructible.trimble.com/construction-industry/what-is-bim-building-information-modeling>>. Accessed 21 September 2019.
- 5 Pitkänen H., Business importance of the given research topic. Finland, Espoo HQ. August 2019.
- 6 Bishop C. M., Pattern Recognition and Machine Learning. Springer. 2006.
- 7 Trimble Inc. About Trimble. Company Description <[https://www.trimble.com/Corporate/About\\_at\\_Glance.aspx](https://www.trimble.com/Corporate/About_at_Glance.aspx)>. Accessed 30 September 2019.
- 8 Trimble Tekla. Powerful structural BIM software for better projects. Product Description <<https://www.tekla.com/products/tekla-structures>>. Accessed 30 September 2019.
- 9 Tekla Corporation, Tekla Structures Steel Connection Guide. 2015.
- 10 Tekla Support. Autoconnections overview. Product Description <[https://teklastructures.support.tekla.com/2019/en/sys\\_autoconnections\\_overview](https://teklastructures.support.tekla.com/2019/en/sys_autoconnections_overview)>. Accessed 15 November 2019.
- 11 RedHat. What are Application Programming Interfaces. Internet Article <<https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces>>. Accessed 02 January 2020.
- 12 Tekla Structures. Tekla Open API. Documentation <<https://developer.tekla.com/tekla-structures/tekla-open-api>>. Accessed 02 January 2020.
- 13 James G., Witten D., Hastie T., Tibshirani R., An Introduction to Statistical Learning. Springer. 2013, pp. 204.
- 14 Lei Y., Huan L., Feature selection for high-dimensional data: a fast correlation-based filter solution. 2003.
- 15 Davis J.A., Clustering and structural balance in graphs. 1967.
- 16 Hartuv E., Shamir R., A clustering algorithm based on graph connectivity. December 2000.
- 17 Xu X., Ester M., Kriegel H.P., A distribution-based clustering algorithm for mining in large spatial databases. 2013.

- 18 Kriegel H., Kröger P., Sander J., Zimek A., Density-based Clustering. 2011.
- 19 Huang, Z., Extensions to the k-means algorithm for clustering large data sets with categorical values. 1998.
- 20 Shyam B., Varun C., Vipin K., Similarity Measures for Categorical Data: A Comparative Evaluation. 2008.
- 21 Anton H., Elementary Linear Algebra, Seventh Edition. John Wiley & Sons. 1994, pp. 170–171.
- 22 Krause E.F., Taxicab Geometry. 1987.
- 23 Jiawei H., Jian P., Data Mining, Third Edition. 2012.
- 24 Tan P.N., Steinbach M., Kumar V., Introduction to Data Mining, Addison-Wesley. 2005, pp.500.
- 25 Hancock J.M., Jaccard distance (Jaccard Index, Jaccard Similarity Coefficient). 2004
- 26 Rokach L. M., Data mining with decision trees: theory and applications, World Scientific Pub Co Inc. 2008.
- 27 Schmidhuber J., Deep Learning in Neural Networks: An Overview. 2015.
- 28 Pitkänen H., Analysis of prototype outcomes with a specialist. Finland, Espoo HQ. February 2020.